

1005728-012502

- 1 1. A method of mapping a plurality of tasks and data onto a multiple processor,
2 distributed memory hardware architecture, the method comprising:
 - 3 a) describing a task-level network of behaviors, each of the task-level
4 network of behaviors being related through control and data flow;
 - 5 b) predicting a schedule of tasks for the task-level network of behaviors; and
 - 6 c) allocating the plurality of tasks and data to at least one of the multiple
7 processors and to at least one of distributed memory, respectively, in
8 response to the predicted schedule of tasks.
- 1 2. The method of claim 1 wherein the predicting the schedule of tasks comprises
2 minimizing execution time of the plurality of tasks.
- 1 3. The method of claim 2 wherein the predicting the schedule of tasks comprises
2 minimizing the schedule of tasks by allocating data to the distributed memories in
3 order to minimize data transfers.
- 1 4. The method of claim 1 wherein the predicting the schedule of tasks comprises
2 maximizing parallel execution of the plurality of tasks on at least two processors
3 of the multiple processors.
- 1 5. The method of claim 1 wherein the allocating the plurality of tasks comprises
2 allocating tasks to one of the multiple processors having optimal processor
3 resources for the tasks.
- 1 6. The method of claim 1 wherein the predicting the schedule of tasks comprises

2 using a resource-based model of the architecture to predict the schedule of tasks.

1 7. The method of claim 1 wherein the predicting the schedule of tasks comprises
2 using an interval graph and an execution time model of the task-level network of
3 behaviors to predict the schedule of tasks.

1 8. The method of claim 1 wherein the allocating the plurality of tasks and data
2 comprises an iterative allocation process.

1 9. The method of claim 8 wherein the iterative allocation process comprises using a
2 demand-driven and constraint-based objective function.

1 10. The method of claim 1 wherein the describing a task-level network of behaviors
2 comprises describing a task-level network of behaviors in a high-level
3 programming language.

1 11. The method of claim 10 wherein the describing the task-level network of
2 behaviors in the high-level programming language comprises parsing the high-
3 level programming language into an intermediate form.

1 12. The method of claim 1 further comprising generating machine executable code
2 for the multiple processor, distributed memory hardware architecture based at
3 least in part on the allocating the plurality of tasks and data.

1 13. The method of claim 1 wherein the allocating the plurality of data to the
2 distributed memories comprises allocating data to shared memories.

1 14. The method of claim 1 wherein the allocating the plurality of data to the

2 distributed memories comprises allocating data to private memories.

1 15. A method for generating a control graph for a compiler used to map a plurality of
2 tasks and data onto a multiple processor, distributed memory hardware
3 architecture, the method comprising:

- 4 a) parsing a plurality of tasks into an internal compiler form of
5 interconnected task nodes; and
- 6 b) linking a compiler representation to each interconnected task node using
7 directed edges for the purpose of substantially simultaneously mapping the
8 tasks to multiple processors in the multiple processor, distributed memory
9 hardware architecture.

1 16. The method of claim 15 further comprising binding the interconnected task nodes
2 to the directed edges.

1 17. The method of claim 15 further comprising parsing a plurality of data blocks into
2 an internal compiler form of data nodes and linking a compiler representation to
3 each data node using directed edges.

1 18. The method of claim 15 wherein the directed edges represent time intervals.

1 19. The method of claim 18 wherein the time interval comprises the time period
2 between tasks.

1 20. The method of claim 18 wherein the time interval comprises the time period from
2 beginning a task to ending a task.

1 21. The method of claim 18 wherein the time interval comprises a set time period.

1 22. The method of claim 18 wherein the time interval comprises a time period
2 between time periods.

1 23. A method for executing a schedule of tasks in a multiple processor, distributed
2 memory architecture, the method comprising:

3 a) generating the schedule of tasks based at least in part on a task-level
4 network of behaviors;

5 b) calculating a demand function based at least in part on a constraint related
6 to at least one of a plurality of tasks in the schedule of tasks; and

7 c) allocating a task having highest priority to a processor having least cost
8 according to the demand function.

1 24. The method of claim 23 further comprising allocating a data block to a memory in
2 the distributed memory.

1 25. The method of claim 23 wherein the demand function is calculated based at least
2 in part on the task-level network of behaviors.

1 26. The method of claim 23 wherein the demand function is calculated based at least
2 in part on an impact on the schedule of tasks.

1 27. The method of claim 23 wherein the demand function is calculated based at least
2 in part on an impact on data movement.

- 1 28. The method of claim 23 wherein the demand function is calculated based at least
2 in part on prior allocation decisions.
- 1 29. The method of claim 23 wherein the cost is defined as the least negative impact
2 on at least one performance factor.
- 1 30. The method of claim 29 wherein the at least one performance factor comprises the
2 schedule of tasks.
- 1 31. The method of claim 29 wherein the at least one performance factor comprises
2 data movement.
- 1 32. The method of claim 23 further comprising allocating a task having next highest
2 priority to a processor having next least cost according to the demand function.
- 1 33. The method of claim 23 further comprising recalculating the demand function in
2 response to each task in the plurality of tasks being allocated to a processor.
- 1 34. A compiler for mapping a plurality of tasks and data onto a multiple processor,
2 distributed memory architecture, the compiler comprising:
- 3 a) means for describing a task-level network of behaviors, each of the task-
4 level network of behaviors being interrelated through control and data
5 flow dependencies;
- 6 b) means for predicting a schedule of tasks for the task-level network; and
- 7 c) means for allocating the plurality tasks and data to at least one of the
8 multiple processors and at least one of the distributed memories,

9 respectively, in response to the predicted schedule of tasks.

1 35. The method of claim 34 further comprising means for producing machine
2 executable code for the architecture based at least in part on the means for
3 allocating the plurality of tasks and data.

1 36. The method of claim 34 wherein the multiple processors communicate using the
2 distributed memories.

1005728-91502